

REMARKS

Claims 1-23 are pending. Claims 1, 5, 7-9, 11, 13-15, and 22-23 are herein amended.

The Examiner's acceptance of the originally filed drawings is noted with appreciation.

Claims 1-23 were rejected under 35 U.S.C. § 102(e) as being anticipated by Graham (U.S. Patent Application No. 20030055826).

The Applicants traverse this rejection.

The Applicants' independent **claim 1** defines, in part, a: "system for connecting an application to a database ... comprising: a software driver for connecting said application to said database ...; a distributed transaction coordinator unit residing on said database server ...; a software transaction switch residing on said database server ... for routing said transaction request to the distributed transaction coordinator; and a software library residing on said database server, said software library including at least one procedure for mapping said transaction request to said transaction switch." Independent **claim 8** defines a system having similar limitations.

Graham discloses a system for interfacing multiple application clients with multiple disparate databases via a single, extensible database interface facility 110 and a driver. (Abstract; paragraph #s 0002, 0017, and 0031). The database interface facility 110 is configured to send and receive messages to access databases 112, 114, and 116, and to receive results from operations performed within the database 112, 114, and 116. Database interface facility 110 is further configured to receive a request for a database operation from

clients 104, 106 and 108, and to convert the request into valid database commands, such as a SQL statements. Database interface facility 110 is also configured to determine which database to access based on the request and/or the client. (Paragraph # 0033). The database interface facility 110 includes a database daemon 206 and a proxy 210. The daemon 206 is configured to assign a client 202 to a transaction proxy within transaction pool 210, so as to pass data between the client to the database. (Paragraph # 0036; see also Figure 4, which clearly shows that the proxy 210 is the interface between the database and the client).

Graham further discloses that a transaction proxy 210 may be a JAVA object that operates as a wrapper for a JDBC driver. (Paragraph # 0039). In addition, Graham discloses a statement interpreter 208 that receives a transaction request from the proxy 210, converts the request to a valid database operation (e.g., SQL), and returns the database operation to the proxy 210. (Paragraph #s 0038, 0049-0050; Figure 4, items S410 and S411). Graham further discloses a results interpreter 204 that receives the results from the database operation via the proxy 210, converts the results to usable formatted data, and returns the usable formatted data to the proxy 210. (Paragraph #s 0037, 0052; Figure 4, items S415 and S416).

In order for Graham to anticipate the claimed invention, Graham must recite each and every limitation. As understood by the Applicants, the Examiner suggests that: proxy 210 is the claimed driver; daemon 206 is the claimed distributed transaction coordinator; and statement interpreter 208 is the claimed transaction switch. However, other than referring to the last five lines of Graham's paragraph #0030, the Examiner has not indicated what specific component in Graham discloses or suggests the originally claimed "software library residing on said database server, said software library including at least one procedure for carrying out said transaction request." The Applicants have reviewed Graham in its entirety,

and respectfully submit that Graham fails to disclose or suggest any such library, *in addition to the other elements recited in the claims 1-8*. Moreover, the Applicants have amended the claims 1-8 to more distinctly define the claimed invention, so that the software library includes “at least one procedure for mapping said transaction request to said transaction switch.” In addition, and as explained in more detail with reference to claim 9, Graham is silent on database server architecture.

For at least these reasons, the Applicants respectfully submit that claims 1-8 are patentably distinct over Graham.

Independent **claim 9** defines, in part, a: “method for performing a distributed transaction by connecting an application to a database . . . comprising: making a transaction request . . . to execute a plurality of stored procedures on said database; mapping said transaction request to a transaction switch residing on said database server using at least one of said stored procedures; . . . sending said transaction request, via said transaction switch, to a library in a distributed transaction coordinator residing on said database server . . .”

The Examiner refers to paragraphs #s 0036, 0038, and 0039 as disclosing the limitations recited in claim 9. The Applicants have reviewed Graham and respectfully submit that Graham does not disclose or suggest the combination of mapping a transaction request to a transaction switch residing on said database server using at least one of said stored procedures *and* sending the transaction request, via said transaction switch, to a library in a distributed transaction coordinator residing on said database server. Furthermore, note that Graham discloses that the “daemon 206 is a Java servant” (Paragraph #s 0027, 0036) and further expressly defines a servant as “a modular piece of code, which runs within an

application server system framework." (Paragraph #s 0028). Thus, the Applicants further submit that Graham's interface facility 110 operates in front of (or as part of) the driver (proxy 210) (see Paragraph #0029: "A transaction proxy may be a JAVA object that operates as a wrapper for a JDBC driver"). In this sense, Graham is silent as to database server architecture, and fails to disclose or suggest a "transaction switch residing on said database server" and a "library in a distributed transaction coordinator residing on said database server" as recited in independent claim 9 and its dependent claims 10-14. In addition, the Applicants have amended independent claim 9 to more distinctly define the claimed invention.

For at least these reasons, the Applicants respectfully submit that claims 9-14 are patentably distinct over Graham.

Independent **claim 15** defines, in part, a: "method for performing a distributed transaction by connecting an application to a database ..., comprising: transmitting a start request ...; calling an extended stored procedure ... stored in said database; calling, by the extended stored procedure, a start function in a transaction switch library residing on said database server; obtaining, by the extended stored procedure, a transaction ID from said database and returning said transaction ID to a said application server; and sending a data packet to said database along with the transaction ID instructing the database server to enlist a specified transaction in a distributed transaction."

The significance of Graham's silence as to database server architecture is equally applicable here, as Graham fails to disclose or suggest a "calling, by the extended stored procedure, a start function in a transaction switch library residing on said database server"

and “sending a data packet to said database along with the transaction ID instructing the database server to enlist a specified transaction in a distributed transaction” as recited in independent claim 15. Moreover, the Examiner refers to the results interpreter 204 as disclosing: “calling, by the extended stored procedure, a start function in a transaction switch library residing on said database server.” The Applicants respectfully submit that the results interpreter 204 merely receives results from a database operation via a proxy, and converts those results into usable formatted data based on client 202 (see, for example, Paragraph # 0037 and Figure 4, items S414 through S416). Thus, Graham is not concerned with how the database operation is carried out; rather, Graham assumes the database operation takes place conventionally (at some database server operation) and then the results from that operation are returned to the interface 110 of the application server framework. In contrast, the claimed invention is expressly involved in carrying out the database operation in that the extended stored procedure calls a “start function in a transaction switch library” residing on the database server. In addition, the Applicants have amended claim 15 to more distinctly define the claimed invention.

For at least these reasons, the Applicants respectfully submit that the claims 15-23 are patentably distinct over the cited art.

Also, the Applicants do not concede that Graham has an effective filing date that is prior to the Applicants’ date of invention, and the Applicants reserve the right to swear behind Graham in accordance with Rule 131.

Based on the above remarks, the Applicants respectfully request the Examiner to reconsider and withdraw the rejection of claims 1-23. Favorable action is solicited. The

Applicants kindly invite the Examiner to contact the undersigned attorney by telephone, facsimile, or email for quickest resolution, if there are any remaining issues.

Respectfully submitted,
JUN CHEN, ET AL.

Dated: April 10, 2006

By: /Neil F. Maloney, Reg. No. 42,833/
Neil F. Maloney, Reg. No.: 42,833
Fenwick & West LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94041
Tel.: (650) 335-7127
Fax.: (650) 938-5200